Git Sourcetree 视图工具

作者:孙元清 Wx：SunYusnQing8

前言：相信各位在技术圈至少也有了一两年了，知道这个是干嘛的，本教程已经包含常用的 git 功能，如果觉得有帮助到在开发中的你，请给这个项目点赞，开始安装使用过程个别 PC 必然会有一些奇葩问题可以找我，不是本公司需要一点小费。

0. 它是一个怎样的工具?

它是和 git 深度融合的一个 git 版本管理的视图工具,当然它和终端命令行没有什么区别,一个是视图,一个是命令行而已. 它和终端执行的命令是同步的一体的.

1.界面工作区面板介绍

小提示它要账号登录才能使用，没有别的用途,所以拿笔记好,怕以后忘了.

账号登录的平台在国外...

设置 git 仓库的账号的用户

查看当前用户名和邮箱

```
git config user.name
git config user.email
```

修改当前用户名和邮箱

```
git config --global user.name "你的名字"
git config --global user.email "你的邮箱"
```

~~设置公钥 ssh-keygen -t rsa -C "你的邮箱"~~

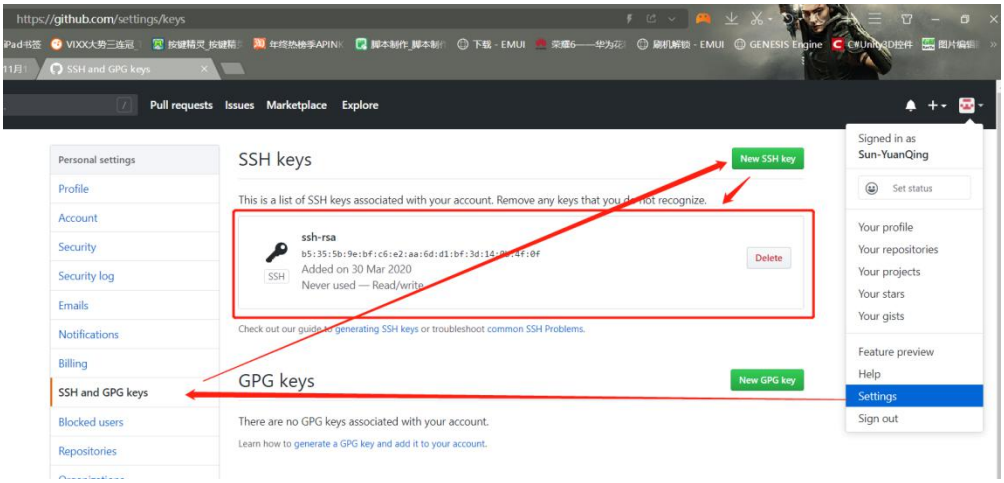
然后输入生产 ssh key 的命令：

ssh-keygen ~~或 ssh-keygen -t rsa -C "你的邮箱"~~

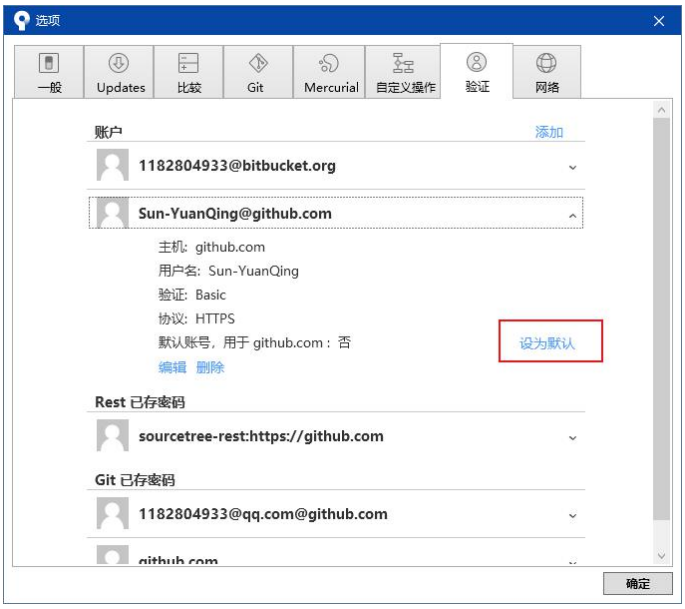
```
Administrator@PC-20200428GAFY MINGW32 /d/my-demo/Century-Cloud-Merchants (SunYua
nQing)
$ git config user.name
sunyuanqing

Administrator@PC-20200428GAFY MINGW32 /d/my-demo/Century-Cloud-Merchants (SunYuanQing)
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Administrator/.ssh/id_rsa):
```

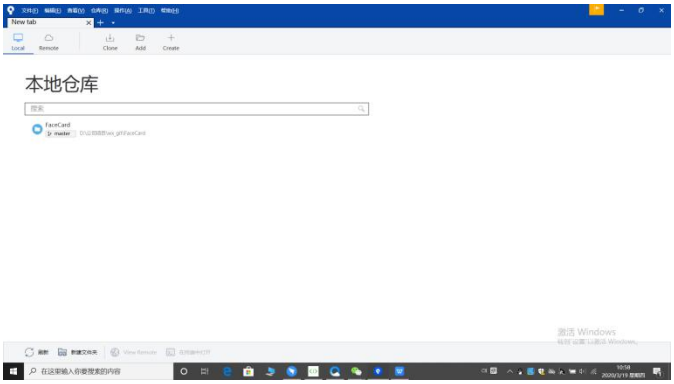
笔记本打开 c/Users/lingh/.ssh/id_rsa.pub.复制内容到 GitHub



工具-->选项

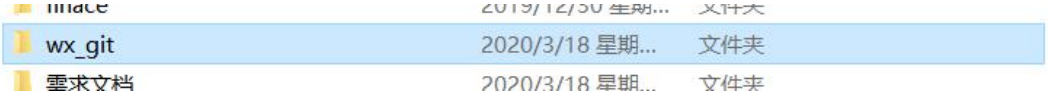


1.1 安装 Sourcetree 并设置用户后初始界面是这样的(我已经有了一个包)

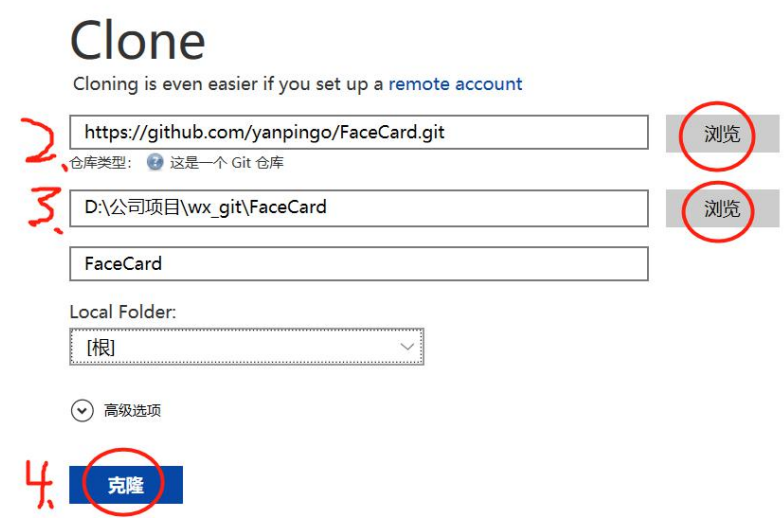


2 从 gitgithub 上克隆一个项目共同开发

我先建一个文件用来放置 git 包源码

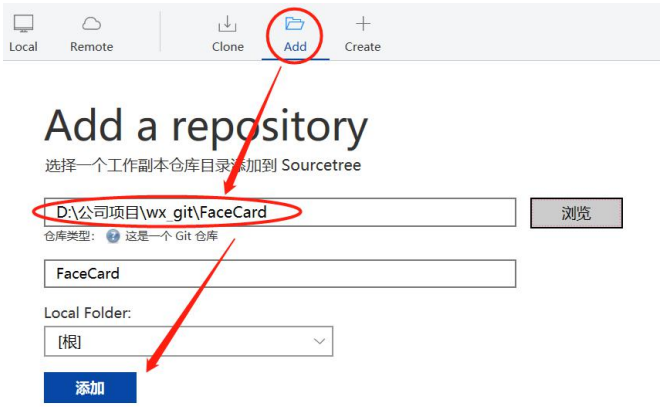


从 gitgithub 上克隆的目标仓库，并指定放到那里.



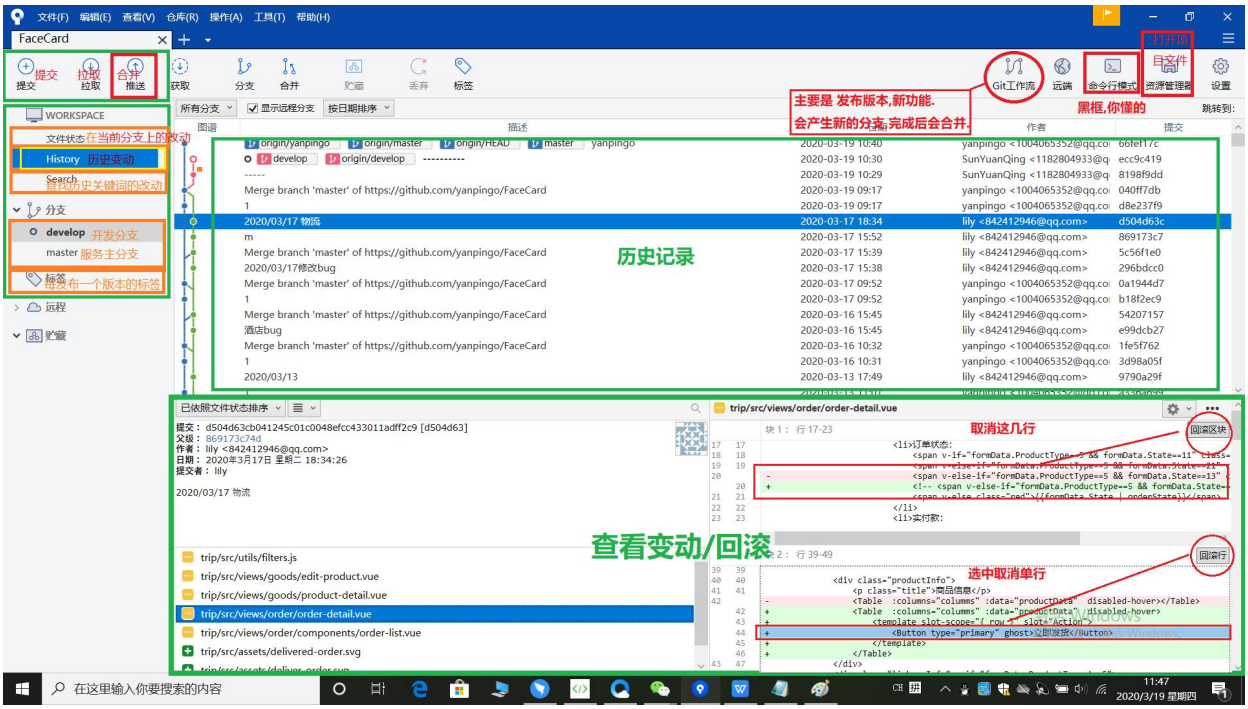
3. 导入现有的本地 GIT 仓库

由于之前就有了 GIT 仓库，再用这款软件照样可以同步，选择 Add 添加本地仓库就好了.



4. 克隆完成，界面工作区的介绍

这个是在 develop 分支下 查看 History 项目 历史变动 的记录列表



5. 分支介绍

5.1. Master

它是主干，相当于一颗大树的主干，一般我们不会在这上面直接修改.

所有子分支 新功能的分支，新版本的分支包含 develop 最终都会合并或推送到 Master 上，形成一个完整的版本.

5.2. Develop

0. 如果没有 Develop 可以通过 Git 工作流初始化产生 Develop.

1. 它是我们开发时的一个分支，最新的代码在这里，这是约定俗成的，新同志可以拉取 Develop 上开发

2. 可能还有基于它的新分支

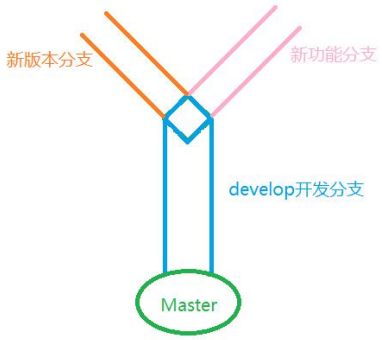
2.1. 新功能(每个人可能 1 个分支，完成新功能后合并. 后面会介绍 新功能分支 的操作)

2.2. 新版本(版本发布者操作的分支，如果你有幸看到，说明发布者正在发布新的项目版本，随后会合并代码到 Develop，Develop 再推送到 Master. 后面会介绍 发布新版本 的操作)

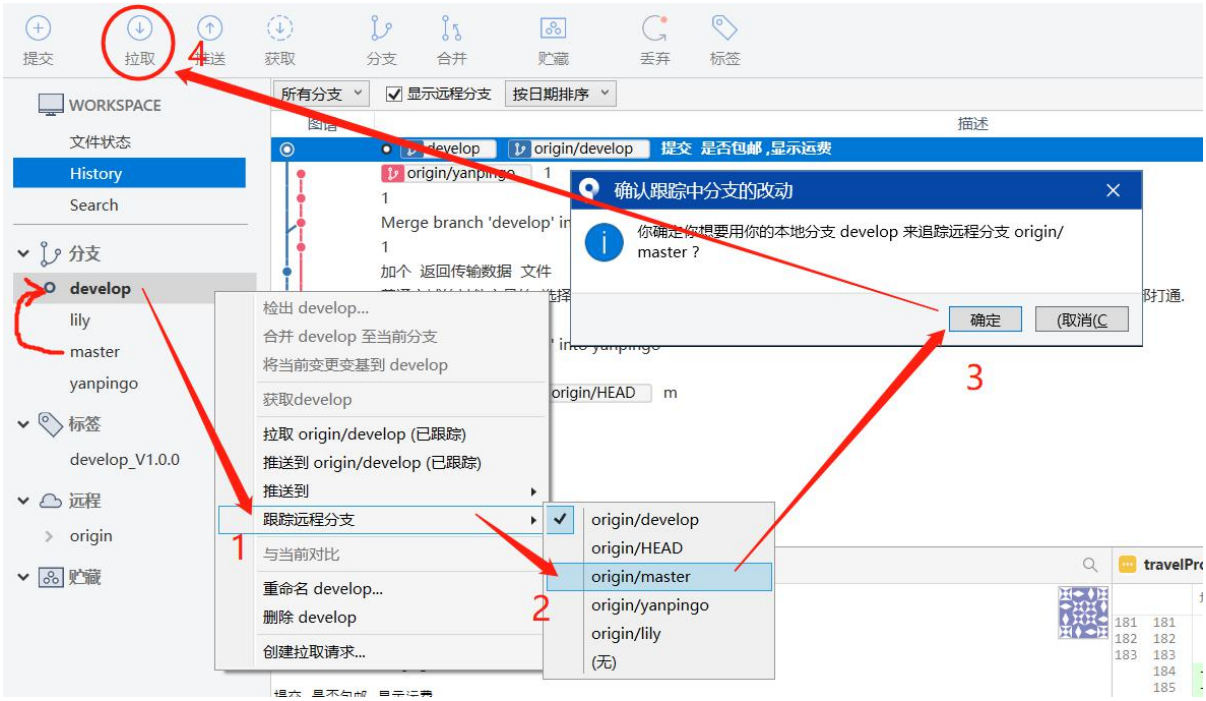
5.3. 其它分支

5.4. 理想分支(master 和 develop)

我们的分支,master,develop,yanpingo,lily 处在 master 同一级



5.5. 【同步合并】其它分支(Master)代码到本分支

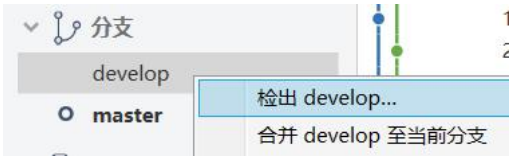


6. 简单的 拉取 ， 提交

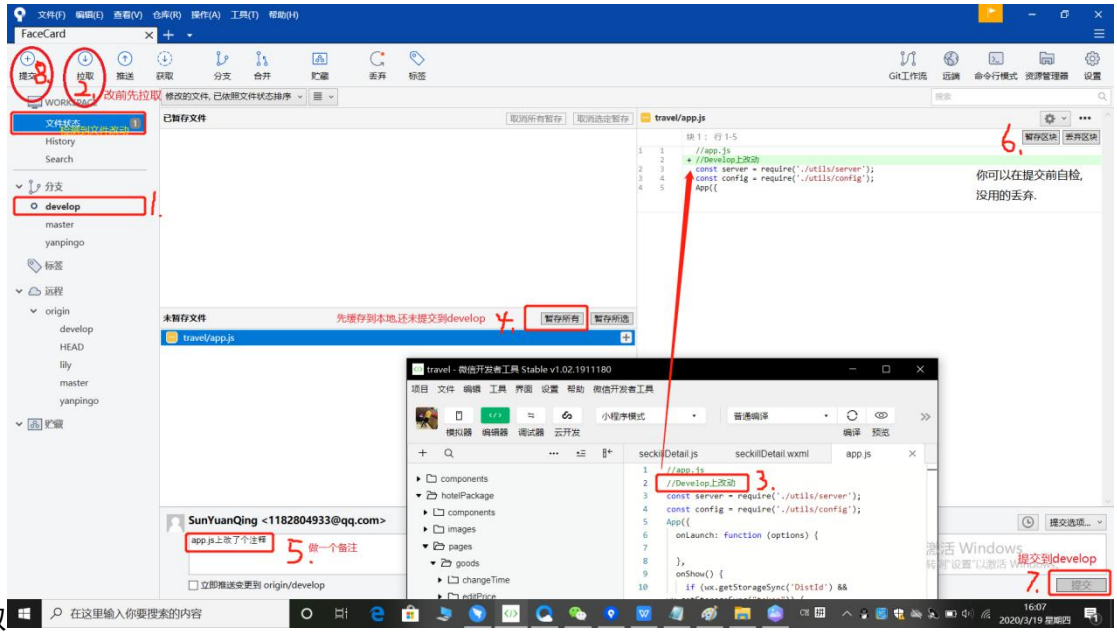
6.1 首先我们将分支切换到 Develop 开发分支上,一切新的改动都基于它.

Develop 右击检出 develop

注意如果 master 上有未提交的文件,检出到其他分支会报错,你要么提交要么丢弃更改.



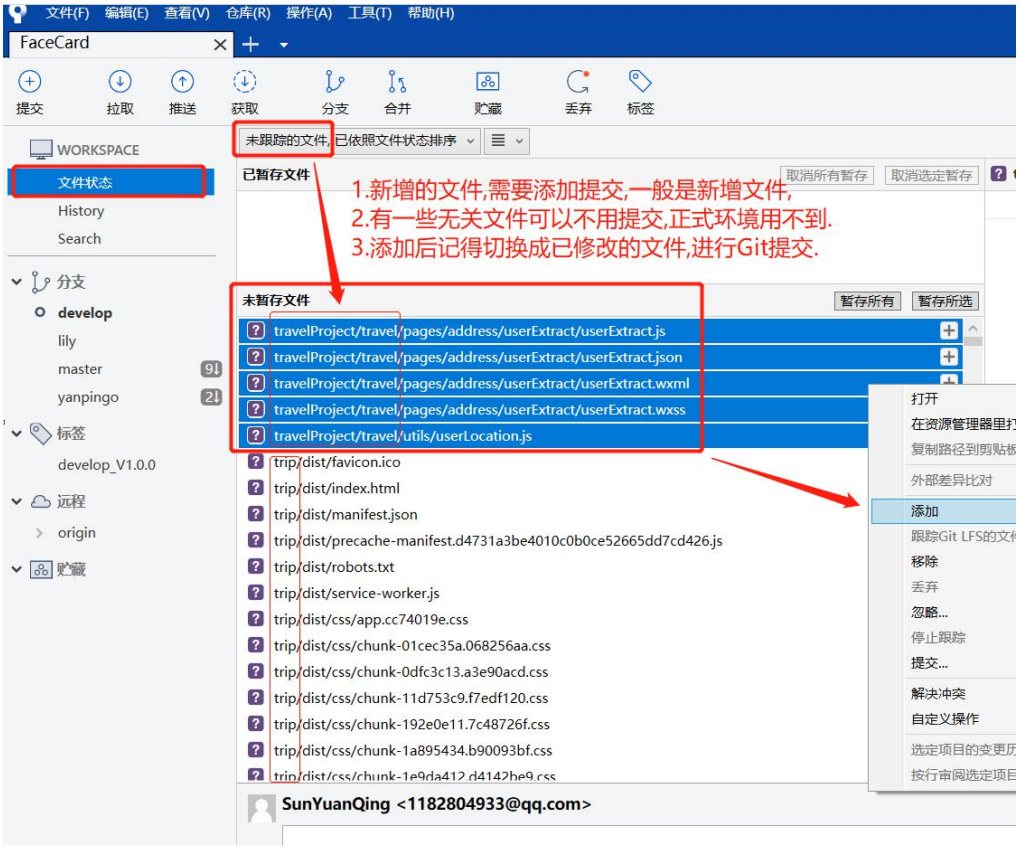
6. 2 提交, 拉取



6. 2. 1 改动之前先拉取

6. 2. 2 新增的文件, 需要添加提交

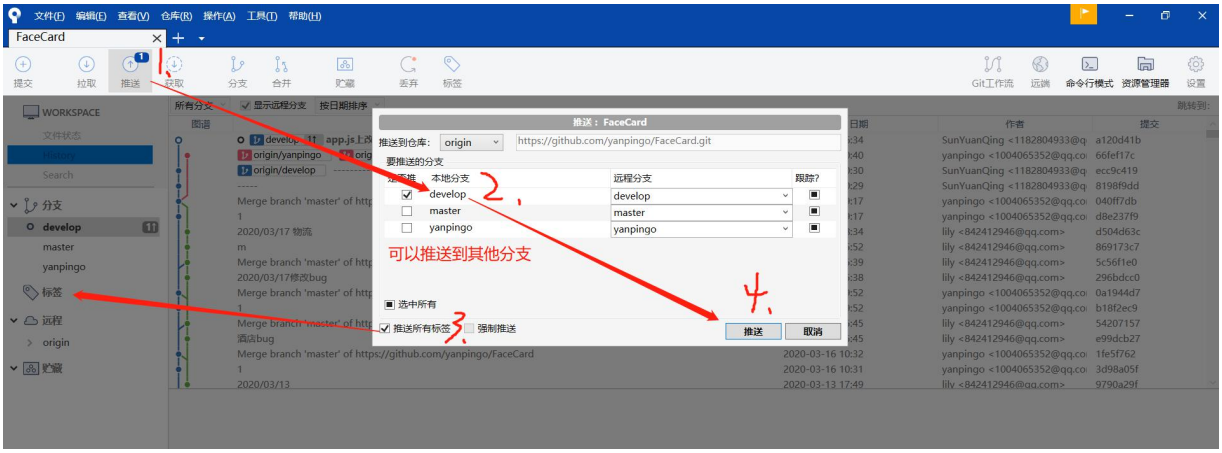
不知道是我设置了还是怎么的, 新增的文件, 需要添加提交, 有一些无关文件可以不用提交. . . . .



6. 3 推送

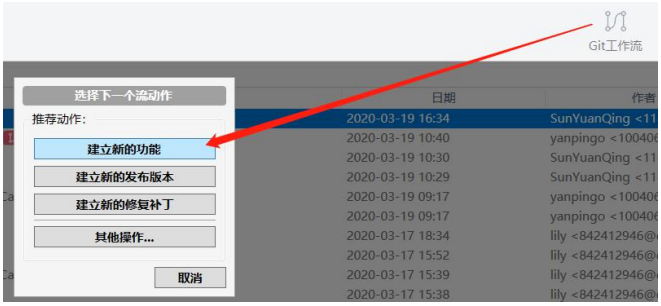
6. 3. 1 推送是将本分支代码给其它分支

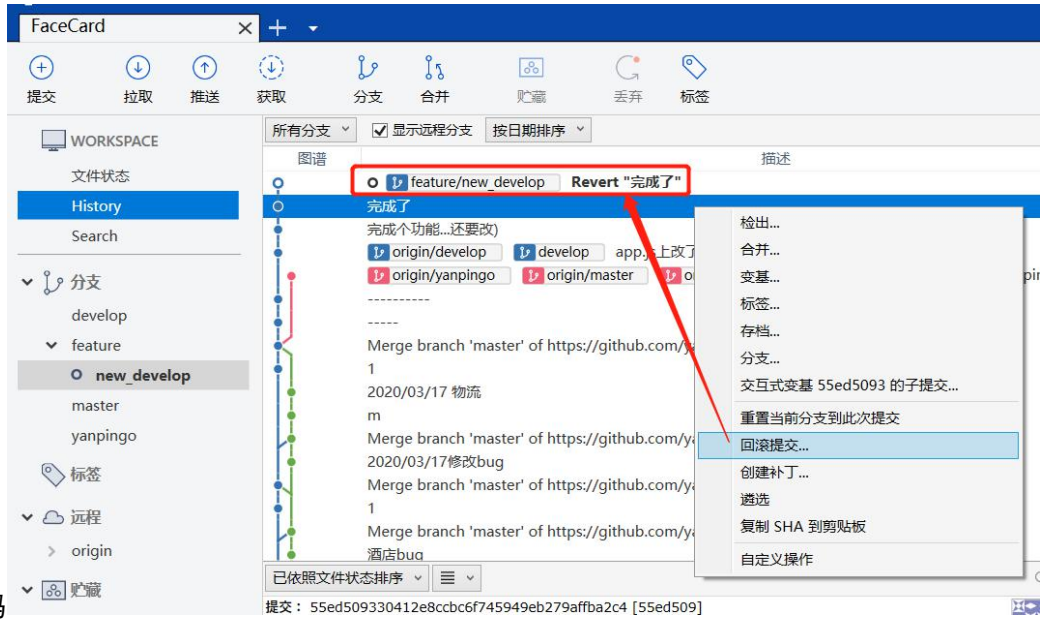
6. 3. 2 分支太多了, 所以记住先切换到所有的分支都拉取一遍, 再推送, 不然可能报错, 重要, 重要, 重要!!!!!



6. [新功能分支]

6.1. Git 工作流---->建立新的功能

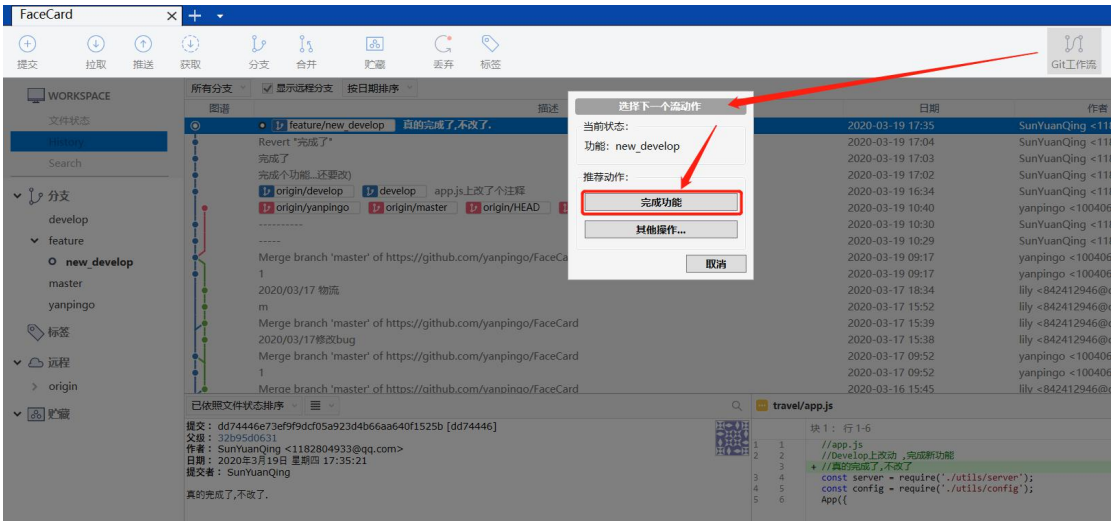




你可以回滚之前提交的代码

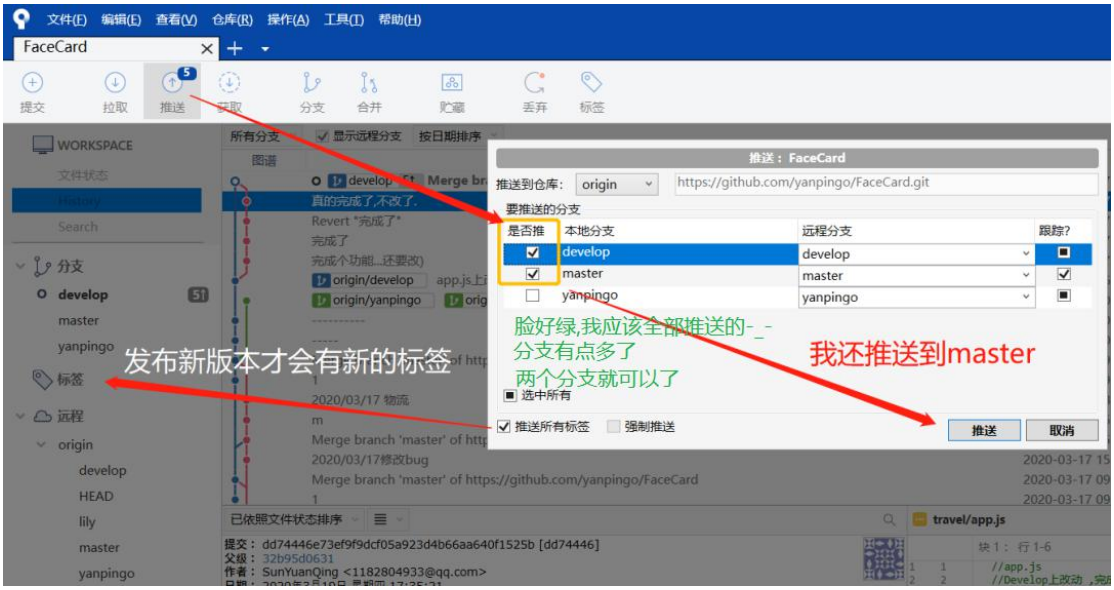
6.2. 经历过一系列代码改动提交完毕以后，选择

6.3. git 工作流--->完成功能



推送是将本分支代码给其它分支,记住先切换到所有的分支都拉取同步合并分支一遍(可能要测试 Master 先不动，等发布新版本同步合并），再推送, 不然可能报错, 重要, 重要, 重要!!!

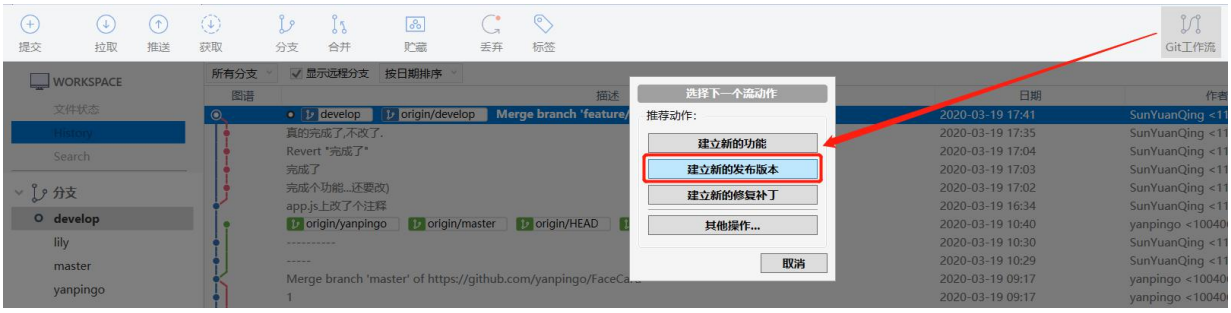
6. 4 再推送



好新功能就介绍完毕了

7. [新版本分支]

7.1. Git 工作流--->建立新的发布版本

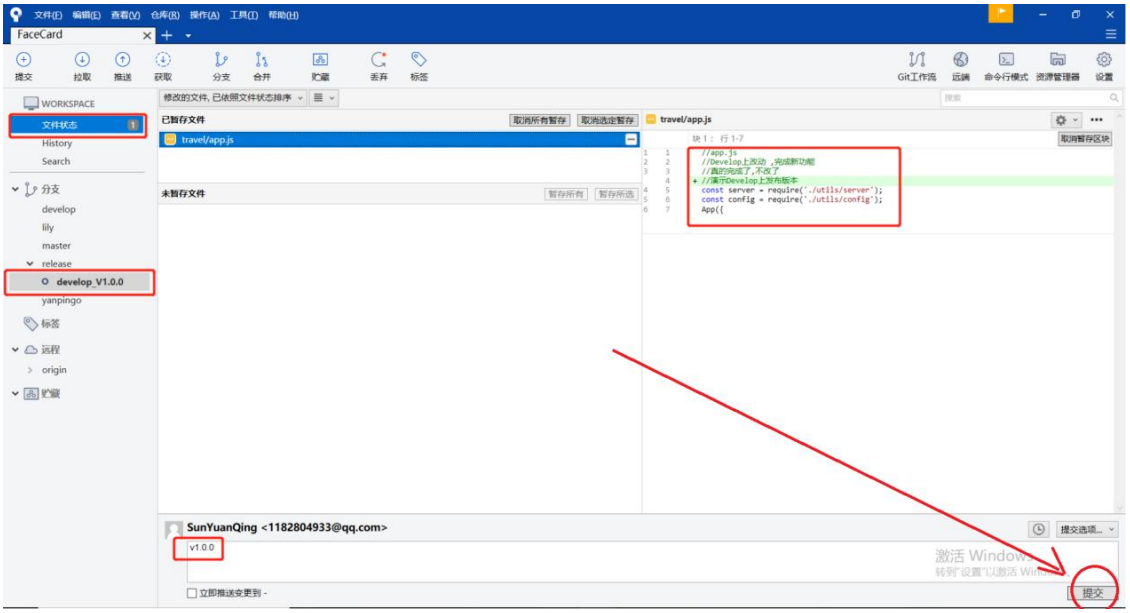


7.2. 启一个版本号 V1.0.0, 确定

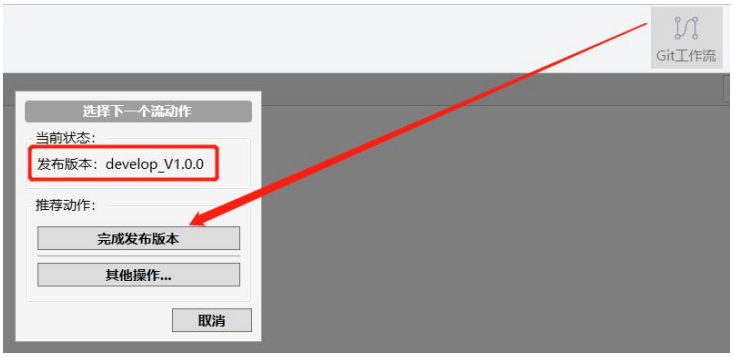
忘了截图...

7.3. 改点东西--->暂存所有

7.4 提交



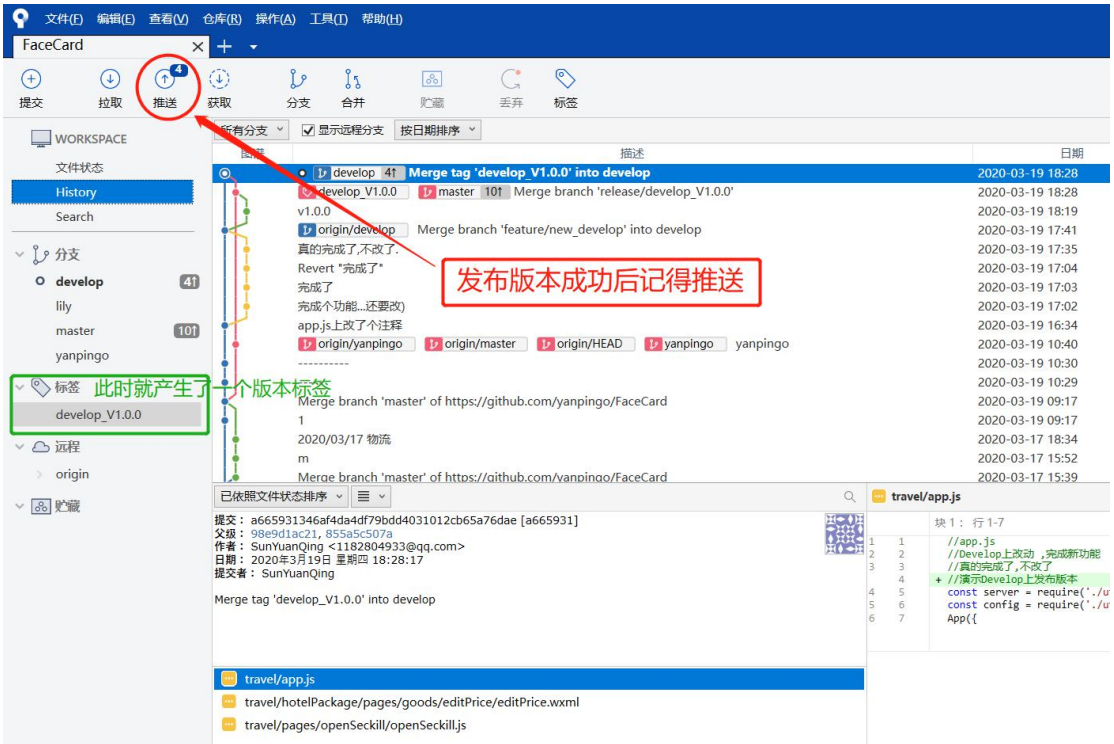
7.5. Git 工作流--->完成版本发布



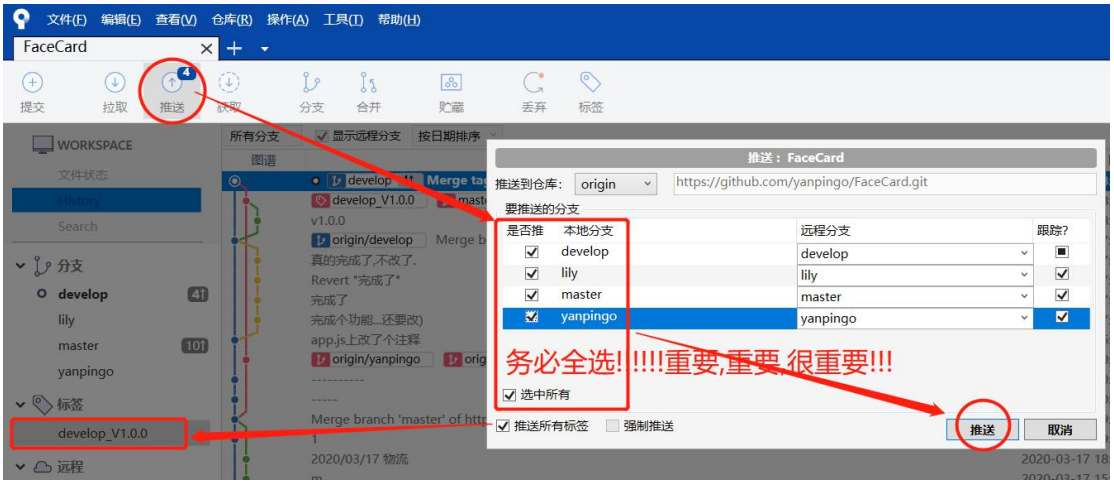
7. 6. 确定

7. 7. 分支太多了, 所以记住先切换到所有的分支都拉取【同步合并】分支一遍, 再推送, 不然可能报错, 重要, 重要, 重要!!!!!

7. 8. 推送



所有分支都


文件(F) 编辑(E) 查看(V) 仓库(R) 操作(A) 工具(T) 帮助(H)
FaceCard
提交
拉取
推送
获取
分支
合并
贮藏
丢弃
标签
WORKSPACE
文件状态
History
Search
分支
develop
lily
master
yanpingo
标签
develop_V1.0.0
远程
所有分支
显示远程分支
按日期排序
图谱
develop_V1.0.0
v1.0.0
origin/develop
真的完成了,不改了.
Revert "完成了"
完成了
完成个功能...还要改)
app.js上改了个注释
origin/yanpingo
Merge branch 'master' of http
2020/03/17 物流
推送：FaceCard
推送到仓库:
origin
https://github.com/yanpingo/FaceCard.git
要推送的分支
是否推
本地分支
远程分支
跟踪?
develop
lily
master
yanpingo
务必全选!!!!!重要,重要,很重要!!!
选中所有
推送所有标签
强制推送
推送
取消